General Computer Science (CH08-320101) Fall 2016

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

November 30, 2016

Contents

1 Assignment 1 (Elementary Math and Induction) – Given Sep. 15., Due Sep. 21.

Problem 1.1 (Unary Natural Numbers)
Let $\oplus$ be the addition operation and $\odot$ be the multiplication operation on unary natural numbers as defined on the slides. Prove or refute that: 35pt

1. $a \oplus b = b \oplus a$
2. $(a \oplus b) \odot c = a \odot c \oplus b \odot c$
3. $a \odot b = b \odot a$

Solution:

1. **Proof**: We proceed by induction over b:

   P.1 Base case:

   $$a \oplus 0 = 0 \oplus a$$

   P.1 W e have $a \oplus 0 = a$ by the addition axiom, so we need $a = 0 \oplus a$. This is easily proven by induction over a.

   P.3 Step case: We first prove by induction over a that

   $$s(b) \oplus a = b \oplus s(a)$$

   P.3 B ase case: $a = 0$. We need $s(b) \oplus 0 = b \oplus s(0)$. But from the previous step we know:

   $$s(b) \oplus 0 = 0 \oplus s(b) = s(b \oplus 0) = b \oplus s(0)$$

   Step case, assume $s(b) \oplus a = b \oplus s(a)$. Now:

   $$s(b) \oplus s(a) = s(s(b) \oplus a) = s(b \oplus s(a)) = b \oplus s(s(a))$$

   Back to the problem, assume $a \oplus b = b \oplus a$ and let's prove that $a \oplus s(b) = s(b) \oplus a$.

   P.4 W e have $a \oplus s(b) = s(a \oplus b) = s(b \oplus a) = b \oplus s(a)$. This is equal to $s(b) \oplus a$, which is what we need to prove. □

2. **Proof**: We proceed by induction over c.

   P.1 Base case: $(a \oplus b) \odot 0 = a \odot 0 \oplus b \odot 0$.

   P.1 T his is true by the fact that $\forall n.n \odot 0 = 0$.

   P.3 Step case: Assume that $(a \oplus b) \odot c = a \odot c \oplus b \odot c$. We need $(a \oplus b) \odot s(c) = a \odot s(c) \oplus b \odot s(c)$

   P.3 W e have, from the multiplication axioms:

   $$\begin{aligned}(a \oplus b) \odot s(c) = (a \oplus b) \oplus (a \oplus b) \odot c = (a \oplus b) \oplus (a \odot c \oplus b \odot c) = \\ = (a \oplus a \odot c) \oplus (b \oplus b \odot c) = a \odot s(c) \oplus b \odot s(c)\end{aligned}$$

   □

3. **Proof**: We proceed by induction over b.

   P.1 Base case: $a \odot 0 = 0 \odot a$ can be easily proven by induciton over a.

   P.2 Step case, assume $a \odot b = b \odot a$. We need $a \odot s(b) = s(b) \odot a$.

   P.2 W e have:

$$a \odot s(b) = a \oplus a \odot b = s(0) \odot a \oplus b \odot a =$$
$$= (s(0) \oplus b) \odot a = (b \oplus s(0)) \odot a = s(b) \odot a$$

   where we used the fact that $a = s(0) \odot a$, something that can be quickly proven by induction over a. □

Problem 1.2 (Induction)

Prove by induction or refute that for all natural numbers n the following assertion holds: $n^3 + 5n$ is divisible by 6. 30pt

---

Solution:

Assertion: $\forall\, n \in \mathbb{N}.\, \frac{n^3+5n}{6}$

Induction:

Base case: $n = 0$ $\qquad \frac{0^3+5\times 0}{6} = 0$

Step case: $n \neq 0$

Induction hypothesis $\qquad \forall\, n \in \mathbb{N}.\, \frac{n^3+5n}{6}$

Induction assertion $\qquad \forall\, n \in \mathbb{N}.\, \frac{n+1^3+5(n+1)}{6}$

$$\frac{n+1^3+5(n+1)}{6} = \frac{(n^2+2n+1^2\times(n+1))+5n+5}{6} =$$

$$\frac{n^3+n^2+2n^2+2n+n+1+5n+5}{6} = \frac{n^3+3n^2+8n+6}{6} =$$

$$\underbrace{\frac{n^3+5n}{6}}_{I.H.} + \frac{3n^2+3n+6}{6}$$

Now we have to prove that the second summand is also divisible by 6

$$\frac{3n^2+3n+6}{6} = \frac{3\times(n^2+n+2)}{6} = \frac{(n^2+n+2)}{2}$$

The last term is even, for the following reasons:
* The product/sum of even numbers is even.
* The product of odd numbers is odd, but the sum of two odd numbers is even. □

An alternative and more elegant solution:
Base cases:
1. $n = 0 : n^3 + 5n = 0$ which is divisible by 6

2. $n = 1 : n^3 + 5n = 6$ which is also divisible by 6

Step case from n to $n + 2$:

$$n + 2^3 + 5(n + 2) = (n^3 + 5n) + 6(n^2 + 2n + 1)$$

This term is divsible by 6 since the first summand is by the induction hypothesis and the second because of the factor 6.

Problem 1.3 (A function on natural numbers)

Figure out the operation η on unary natural numbers defined by the following equations: 15pt

$$\eta(o) = o$$

$$\eta(s(o)) = o$$

$$\eta(s(s(n))) = s(\eta(n))$$

Solution:

The function η halves its argument.

Problem 1.4 (A wrong induction proof)

What is wrong with the following "proof by induction"? 20pt

> **Theorem:** All students of Jacobs University have the same hair color.
>
> **Proof:** We prove the assertion by induction over the number n of students at Jacobs University.
>
> **base case:** $n = 1$. If there is only one student at Jacobs University, then the assertion is obviously true.
>
> **step case:** $n>1$. We assume that the assertion is true for all sets of n students and show that it holds for sets of $n + 1$ students. So let us take a set S of $n + 1$ students. As $n>1$, we can choose students $s \in S$ and $t \in S$ with $s \neq t$ and consider sets $S_s = S\backslash\{s\}$ and $S_t := S\backslash\{t\}$. Clearly, $\#(S_s) = \#(S_t) = n$, so all students in S_s and have the same hair-color by inductive hypothesis, and the same holds for S_t. But $S = S_s \cup S_t$, so any $u \in S$ has the same hair color as the students in $S_s \cap S_t$, which have the same hair color as s and t, and thus all students in S have the same hair color □

Solution:

The problem with the proof is that the inductive step should also cover the case when $n = 1$, which it doesn't. The argument relies on the fact that there intersection of S_s and S_t is non-empty, giving a mediating element that has the same hair color as s and t. But for $n = 1$, $S = \{s, t\}$, and $S_s = \{t\}$, and $S_t = \{s\}$, so $S_s \cap S_t = \emptyset$.

2 Assignment 2 (Elementary Math and Induction) – Given Sep. 11., Due Sep. 28.

Problem 2.1 (Properties of Relations)
Let R and S be (non-empty) relations on some given set A. Prove or refute each of the four statements 25pt

1. If R is reflexive then R^{-1} is reflexive
2. If R and S are transitive then $R \cup S$ is transitive
3. If R is symmetric then all subsets of R are symmetric
4. $R \cup R^{-1}$ is symmetric.

Solution:

1. R is reflexive implies that for all $a \in A$ we have aRa. Since $R^{-1} = \{(x, y) \mid (y, x) \in R\}$ it follows that for all $a \in A$ we have $aR^{-1}a$.
2. Consider the following counterexample:
   Take $a, b, c \in A$ arbitrary and distinct and let $R = \{(a, b)\}$, $S = \{(b, c)\}$. Then, although both R and S are transitive their union is not.
3. Consider the following counterexample:
   Take $a, b \in A$ arbitrary and distinct. Then, although $R = \{(a, b), (b, a)\}$ is symmetric its subset $\{(a, b)\}$ for example is not symmetric.
4. If aRb then $bR^{-1}a$ thus for all $a, b \in A$ such that $(a, b) \in (R \cup R^{-1})$ we also have $(b, a) \in (R \cup R^{-1})$. So the affirmation is true.

Problem 2.2 (Properties of Function Composition)
Let $f \subseteq A \times B$ and $g \subseteq B \times C$ be functions. Prove or refute the following statements: 35pt

1. $g \circ f$ is a function.
2. if f and g are both injective/surjective/bijective, then so is $g \circ f$.
3. $f \circ g$ is also a function and $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.
4. If $f \circ g = \lambda x.x$, then $f = g^{-1}$.

Note: By "refute" we mean "exhibit a counterexample to this claim". Try to make suggestions how the claim can be salvaged, and prove that they work.

Solution:

1. To prove this, we have to show that for all given $a \in A$, there is a unique $c \in C$, such that $g \circ f(a) = c$. Now, using that f is a function, there is a unique $b \in B$, such that $f(a) = b$, and since g is a function, there is a unique $c \in C$, such that $f(b) = c$. Thus $g \circ f(a) = g(f(a)) = g(b) = c$ is unique.
2. To show that $g \circ f$ is injective we choose $g \circ f(a) = f(g(a)) = f(g(a')) = g \circ f(a')$. As f is injective, we have to have $g(a) = g(a')$ and thus (since g is injective too) $a = a'$, which proves the assertion.
   To show that $g \circ f$ is surjective choose some $c \in C$ and show that it is a pre-image in A. As f is surjective there is a $b \in B$ with $f(b) = c$, and (as g is surjective too), there is an a with $g(a) = b$, so $c = f(g(a)) = g \circ f(a)$.
   The the case for bijectivity is proven by combining the two assertions above.
3. $f \circ g$ cannot be a function in general, since $A \neq C$. If $A = C$, then functionhood can be shown just like in case 1.

The second conjecture is incorrect. First, even we need $A = C$ for the functions to make sense.

Take for instance $g = \lambda x \in B.c$, where $c \in C$ is arbitrary, then $f \circ g = \lambda x \in A.c$, which is not injective, so it cannot be bijective if $\#(A) \geq 2$. The correct version would be: If $A = C$ and f and g are both bijective, then $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

4. Note that since $\lambda x \in A.x \colon A \to C$, we have $A \subseteq C$. We have to show that for all $a \in A$, $f(a) = g^{-1}(a)$.

Problem 2.3 Check whether the following are equivalence relations (you can assume all of them are from the set of natural numbers to itself): 20pt

- $x \sim y$ iff $x \equiv y \bmod 3$ for all $x, y \in \mathbb{N}$
- $x \sim y$ iff $x^2 = y^2$ for all $x, y \in \mathbb{N}$
- $x \sim y$ iff $x|y$ (x is a divisor of y).

Solution:

- $\sim$ is an equivalence relation. It is reflexive because $x \equiv x \bmod 3$) for all $x \in \mathbb{N}$. It is also symmetric because if $x \equiv y \bmod 3$ then $y \equiv x \bmod 3$. Transitivity follows from $x \equiv y \bmod 3 \wedge y \equiv z \bmod 3 \implies x \equiv z \bmod 3$.
- $\sim$ is an equivalence relation. $x^2 = x^2$ shows reflexivity, while $x^2 = y^2 \implies y^2 = x^2$ and $x^2 = y^2 \wedge y^2 = z^2 \implies x^2 = z^2$ due to positivity prove symmetry and transitivity.
- $\sim$ is not an equivalence relation since it isn't symmetric. For example, $3|6$ but $\neg 6|3$.

Problem 2.4 (Relation Properties)
You are given the set $A = \{a, b, c, d, e\}$. Find the number of relations $R \subseteq A \times A$, which are simultaneously reflexive and symmetric. 20pt

---

Solution: There are $5 \times 5 = 25$ pairs in A.

R is a reflexive relation. Therefore the pairs (a, a), (b, b), (c, c), (d, d) and (e, e) should all be included in the relation.

R should also be a symmetric relation. Therefore, if (x, y) is included, (y, x) also has to be included. There are $\frac{5 \times 4}{2} = 10$ such couples of pairs. We can choose to include any subset of those in the relation, and it will be symmetric. Thus we have $2^{10} = 1024$ options to choose from.

Therefore there are 1024 such relations over the set A in total.

---

3 Assignment 3 (Functions and SML) – Given Sep. 28., Due Oct. 5.

Problem 3.1 (Are bijective functions with composition a group?)
A group is a set G with a binary operation $*\colon G \times G \to G$, obeying the following axioms: 20pt

Closure: G is closed under $*$, i. e. $\forall\, a, b \in G . a * b \in G$

Associativity: $\forall\, a, b, c \in G . (a * b) * c = a * (b * c)$

Identity element: $\exists\, e \in G . \forall\, a \in G . a * e = e * a = a.$

Inverse elements: $\forall\, a \in G . \exists\, a^{-1} \in G . a * a^{-1} = e.$

If, additionally, the following axiom holds, the group is called "commutative" or "Abelian":

Commutativity: $\forall\, a, b \in G . a * b = b * a.$

- Now prove or refute whether the set of all bijective functions $f\colon A \to A$ on a set A with the function composition $\circ$ forms a group.
- Is it commutative?

Solution:

Associativity:

Identity element: the identity function $\lambda x.x$

Inverse elements: f^{-1}, exists due to bijectivity

Commutativity: No, easy counter-example.

Problem 3.2 (Function Property)
Prove or refute: if a mapping of a finite set to itself is injective then it is surjective as well. 20pt

Solution: Let A be a finite set and let $f\colon A \to A$. Let $a \in A$. Define $f^0(a) = a$. Because A is finite, then $f^i(a), 0 \leq i n$ cannot be all distinct if n is large enough. Thus at some point we have $f^r(a) = f^s(a)$, where $s \leq r$. Because f is injective, there is an inverse mapping f^{-1}. $f^{r-s}(a) = f^0(a) = a$ and so $a = f(f^{r-s-1}(a))$, where $f^{r-s-1}(a)$ is in A. Thus f is surjective.

Problem 3.3 (Function Properties)
Consider the following partial functions in $\mathbb{R} \rightharpoonup \mathbb{R}$ 25pt

1. $f(x) = x^2 - x$
2. $g(x) = \frac{1}{5+e^x}$
3. $h(x) = x \cdot \ln(3 + x)$

Your tasks are the following:

1. Are f, g, and h total? If not, determine the biggest set the function is defined on.
2. Are the three functions injective or not? Justify your answer.
3. Write down the expressions for:

(a) $f \circ g$
(b) $(h \circ f)^{-1}$
(c) $g \circ g$

Solution:

1. The domain of f is $\mathbb{R}$ and its codomain is also $\mathbb{R}$. The domain of g is $\mathbb{R}$ and its codomain is $(0, \frac{1}{5})$
2. f is not injective since $f(0) = f(1) = 0$. g is injective since it is a strictly decreasing function. The argument with $g(x_1) = g(x_2)$ resulting in $x_1 = x_2$ may also be used. h is not injective since $\ln(3 + x)$ and x are continuous strictly increasing functions. This means that for some $x \leq 0$ and some $0 \leq y$, $h(x) = h(y)$.
3. (a) $f \circ g(x) = f(\frac{1}{5+\exp x}) = \frac{1}{5+e^x}^2 - \frac{1}{5+e^x}$
   (b) $(h \circ f)^{-1} = f^{-1} \circ h^{-1}$ does not exist since f is not bijective (not injective from previous task).
   (c) $g \circ g = g(\frac{1}{5+\exp x}) = \frac{1}{5+e^{\frac{1}{5+e^x}}}$

Problem 3.4 (Set Operations) 35pt

Consider integer sets in SML as lists of integers. Implement three functions: one that returns the union of two sets of integers, one that returns the intersection of two sets and another one which returns the set difference between them. The functions should have the following signature:

```
val union = fn : int list * int list -> int list
val intersect = fn : int list * int list -> int list
val setdifference = fn : int list * int list -> int list
```

Example:

```
- union([1,2,3],[2,3,4]);
val it = [1,2,3,4] : int list
- intersect([2,3,1],[4,1,7,6]);
val it = [1] : int list
- setdifference([3,1,7,8],[2,1,5,3]);
val it = [7,8] : int list
```

Solution: The approach would be to implement a member function that checks if an element is in a set and then use it for the two functions:

```
fun member (a, []) = false
  | member (a, x::xls) =
if a=x then true
else member(a, xls);

fun union ([], x) = x
  | union (x::ls, y) =
if member(x,y) then union(ls, y)
else x::union(ls, y);

fun intersect ([], _) = []
```

```
  | intersect (a::lsa, lsb) =
if member(a, lsb) then a::intersect(lsa, lsb)
else intersect(lsa, lsb);

fun setdifference ([]:int list, _) = []
  | setdifference (a::lsa, lsb) =
    if member(a, lsb) then setdifference(lsa, lsb)
                      else a::setdifference(lsa, lsb);
```

---

4 Assignment 4 (SML) – Given Oct. 6., Due Oct. 12.

Problem 4.1 Given a sequence a_n we call $a_0 - a_1 + a_2 - a_3 + \ldots$ the alternating sum of the sequence. Write an SML function alternating with type int list -> int that calculates the alternating sum of the list. 20pt

Solution:

```
fun alternating(nil) = 0
  | alternating(a::b) = a − alternating(b);
```

Problem 4.2 (List of anagrams)
Create an SML function that given a list of strings groups all strings that are anagrams of each other together. Two words are called anagrams if they use the same letters the same amount of times. 35pt

Example and function signature:

```
val listAnagrams = fn : string list −> string list list
− listAnagrams ["algorithm", "listen", "logarithm", "something", "silent", "Silent", "lentis"];
val it = [["algorithm","logarithm"],["listen","silent","lentis"],["something"],["Silent"]] : string list list
```

Solution:

```
fun min [a] = a:char
  | min (a::b::c) = if a < b then min(a::c) else min (b::c)

fun sub (a,[]) = []
  | sub (a, b::c) = if a = b then c else b::(sub (a,c))

fun sublist ([],a) = a
  | sublist (a::b, c) = sublist(b,sub(a,c))

fun sort [] = []
  | sort list = min(list)::sort(sub(min(list),list))

fun anagram a b = sort(explode(a))= sort(explode(b))

fun find (a,[]) = nil
  | find (a, b::c) = if anagram a b then b::(find(a,c)) else find(a,c)

fun listAnagrams [] = nil
  | listAnagrams (a::c) = let val list = a::(find(a,c))
                          in list::(listAnagrams(sublist(list,a::c)))
                          end
```

Problem 4.3 (Duplicates)
Write an SML function that removes all duplicate elements from a list. For instance 20pt

```
remove_duplicates([true, true, false]) = [true, false];
remove_duplicates([5,3,12,3,3,2]) = [5,3,12,2];
```

Solution:

```
fun member(a, h::t) = a=h orelse member(a, t)
  | member(_, nil) = false;

fun helper(found_already, h::t) =
if member(h, found_already) then
   helper(found_already, t)
else
h::helper(h::found_already, t)
  | helper(_, nil) = nil;

fun remove_duplicates(l) = helper(nil, l);
```

Problem 4.4 We call the sequence of natural numbers with $a_0 = 0$, $a_1 = 1$, $a_n = a_{n-1} + a_{n-2}$ the Fibonacci sequence. Write an SML function `fib = fn : int -> int` that calculates the nth Fibonacci number. Also, write an SML function `is_fib = fn : int -> bool` that, given a number n, calculates whether n is a fibonacci number or not. You are allowed to define additional functions to help you. 25pt

---

Solution:

```
fun fib(n) = if n<=0 then 0 else if n=1 then 1 else fib(n−1) + fib(n−2)
```

Note: this solution is extremely inefficient. A better one is covered in class.

```
fun is_fib(x) = let
        fun helper(current, value) =
         if fib(current) < value then helper(current + 1, value)
            else if fib(current = value) then true else false;
    in
        helper(0,x)
    end;
```

---

5 Assignment 5 (Datatypes) – Given Oct. 12., Due Oct. 19.

Problem 5.1 (Pearls)

Ariel, the Little Mermaid, was asked by Ursula to give all her beautiful colored pearls in exchange for getting transformed into a real girl. Ariel wanted very much to meet prince Eric in person, but wouldn't give away all her pearls, so she made a deal with Ursula: she were to keep the k most shiny pearls and give away all the others. 25pt

We know that Ariel's pearls were of four types: the carved pearls are not shiny at all and have a "shininess" multiplier of 0; opal-pearls have a multiplier of 2; the faceted pearls have a multiplier of 4; the akoya pearls have the highest multiplier, namely 10. Each pearl also has an intrinsic "shininess" that gets multiplied by the multiplier.

You are given the following SML datatype and function signature and you are asked to return the total value of the "shininess" that Ariel can get:

```
datatype pearl = carved of int | opal of int | faceted of int | akoya of int;
val shiny = fn : pearl list * int -> int;
```

For instance to compute

```
- shiny([carved(2), opal(3), faceted(1), faceted(5), akoya(2), opal(20)], 3);
val it = 80 : int;
```

we first compute the shininess of the five pearls: [0,6,4,20,20,40], and then add up the shininess of the $k = 3$ shiniest ones.

Solution: First idea: sort the pearls, get the k largest and add up their values.

```
fun len([]) = 0 | len (x::L) = 1+len(L);
fun take(L, 0) = [] | take([], _) = [] | take(x::L, n) = x::take(L,n-1);
fun drop(L, 0) = L | drop([], _) = [] | drop(x::L, n) = drop(L, n-1);

datatype pearl=carved of int | opal of int | faceted of int | akoya of int;

fun value(carved(X)) = 0
  | value(opal(X)) = 2*X
  | value(faceted(X)) = 4*X
  | value(akoya(X)) = 10*X;

fun cmp(A, B) = value(A) > value(B);

fun merge([],p,_) = p
  | merge(p,[],_) = p
  | merge(a::p1, b::p2, compare) = if compare(a,b) then a::merge(p1,b::p2,compare) else b::merge(a::p1,p2,c

fun sort([], _) = []
  | sort([a], _) = [a]
  | sort(L,cmp) = let val l1 = len(L) val p1 = sort(take(L,l1 div 2),cmp) val p2 = sort(drop(L,l1 div 2),cmp
in merge(p1,p2,cmp) end;
```

```
fun sum([]) = 0
  | sum(a::L) = value(a) + sum(L);

fun shiny(X,k) = sum(take(sort(X,cmp),k));
```

A nice variant would be to use function intsort that sorts lists of integers and then

```
fun shiny(X,k) = foldl op+ 0 (take(intsort(map value X),k));
```

Third idea: we can obtain the result for a list $a :: L$ and k pearls left by either taking the a element into account: $value(a) + shiny(L, k-1)$ or dropping a: $shiny(L, k)$. The following (shorter) implementation uses the idea:

```
datatype pearl=carved of int | opal of int | faceted of int | akoya of int;

fun value(carved(X)) = 0
  | value(opal(X)) = 2*X
  | value(faceted(X)) = 4*X
  | value(akoya(X)) = 10*X;

fun max(a, b) = if a>b then a else b;

fun shiny([], _) = 0
  | shiny(_, 0) = 0
  | shiny(a::L, k) = max( value(a)+shiny(L,k-1), shiny(L,k) );
```

Problem 5.2 (Music notes) 20pt

In music theory, notes are divided into groups of 12 semitones. Those groups are called octaves. In each octave, we have the familiar 7 note classes - A, B, C, D, E, F and G (ordered from lowest to highest). In order to denote all 12 semitones, we use the special symbols operators: sharp ($\sharp$) and flat ($\flat$), which mean 1 semitone higher and lower, respectively. Now we only need to know the number of semitones between those classes:

A to B : 2 semitones	B to C : 1 semitone
C to D : 2 semitones	D to E : 2 semitones
E to F : 1 semitone	F to G : 2 semitones
G to A : 2 semitones	:

This leads to many different representations of the same note. For example, $C = \sharp B = \flat\flat D = \sharp\flat C$

For a computer program, representing each note with a unique integer is much more convenient. To do that, we number all the notes by increasing pitch: A from octave 0 becomes 0, $\sharp A$ from octave 0 becomes 1, A from octave 1 becomes 12, B from octave 1 become 14, etc.

You are given an SML datatype for notes:

```
datatype noteclass = A | B | C | D | E | F | G |
                   | sharp of noteclass
                   | flat of noteclass;

datatype note = note of int * noteclass; (* octave number and note class *)
```

Write an SML function that converts a list of notes from the datatype to the integer representation described above. Example and signature:

```
val convert = fn : note list -> int list

- convert([note(0,sharp(A)), note(1,flat(flat(B)))]);
val it = [1,12] : int list
```

Solution:

```
fun toneval(note(x,A)) = x*12
  | toneval(note(x,B)) = x*12+2
  | toneval(note(x,C)) = x*12+3
  | toneval(note(x,D)) = x*12+5
  | toneval(note(x,E)) = x*12+7
  | toneval(note(x,F)) = x*12+8
  | toneval(note(x,G)) = x*12+10
  | toneval(note(x,sharp(n))) = toneval(note(x,n)) + 1
  | toneval(note(x,flat(n))) = toneval(note(x,n)) - 1;

fmiun convert(ls) = map toneval ls;
```

Problem 5.3 (Substitutions)
You are given the ADT 15pt

$$\langle\{\mathbb{A},\mathbb{B},\mathbb{C}\},\{[a\colon\mathbb{A}],[b\colon\mathbb{B}],[c\colon\mathbb{C}],[f\colon\mathbb{A}\to\mathbb{A}],[g\colon\mathbb{A}\times\mathbb{B}\to\mathbb{A}],[h\colon\mathbb{B}\times\mathbb{B}\to\mathbb{A}]\}\rangle$$

Which of the following mappings are valid substitutions?

$\sigma_1 := [f(x_\mathbb{A})/x_\mathbb{A}],[c/y_\mathbb{C}],[g(x_\mathbb{A},b)/Z_\mathbb{A}]$
$\sigma_2 := [h(a,b)/x_\mathbb{A}],[g(a,b)/y_\mathbb{A}]$
$\sigma_3 := [f(a,c)/x_\mathbb{A}],[g(a,b)/y_\mathbb{A}]$
$\sigma_4 := [f^{i+1}(x_\mathbb{A})/f^i(x_\mathbb{A})]$, $i\in\mathbb{N}$ with $f^0(x_\mathbb{A})=x_\mathbb{A}$ and $f^{i+1}(x_\mathbb{A})=f(f^i(x_\mathbb{A}))$
Justify your answers.

Solution: σ_1 is valid since all sorts correspond.
σ_2 is not valid since $h(a,b)$ is not a proper term: sorts don't match: $\mathbb{A}\times\mathbb{B}$ instead of $\mathbb{B}\times\mathbb{B}$
σ_3 is not valid since $f(a,c)$ is not a proper term: sorts don't match: $\mathbb{A}\times\mathbb{A}$ instead of $\mathbb{A}$
σ_4 is not valid since it's not a mapping from $\mathcal{V}$ to $\mathcal{T}(\mathcal{A};\mathcal{V})$ (variables to terms).

Problem 5.4 (Apply substitutions)
Apply the substitutions $\sigma := [h(a,b)/x],[g(c,f(c),b)/y]$ and $\tau := [a/x],[b/c],[h(b,c)/y]$ to the terms $s := h(g(x,y,c),x)$ and $t := g(y,h(y,x),a)$ 15pt
(give the 4 result terms $\sigma(s)$, $\sigma(t)$, $\tau(s)$, and $\tau(t)$).

Prove of refute that substitution application is commutative , i.e. whether $[t_1/x^1_\mathbb{A}]([t_2/x^2_\mathbb{A}](s)) = [t_2/x^2_\mathbb{A}]([t_1/x^1_\mathbb{A}](s))$

Solution:

$$\begin{array}{lcllcl}\sigma(s) & = & h(g(h(a,b),g(c,f(c),b),c),h(a,b)) & \sigma(t) & = & g(g(c,f(c),b),h(g(c,f(c),a),h(a,b)),a)\\ \tau(s) & = & h(g(a,h(b,c),b),a) & \tau(t) & = & g(h(b,c),h(h(b,c),a),a)\end{array}$$

Substitution application is not commutative. Proof by conterexample.

Problem 5.5 (Nested lists)

In class, we have defined an abstract data type for lists of natural numbers. Using this intuition, construct an abstract data type for lists that contain natural numbers or lists (nested up to arbitrary depth). Give the constructor term (the trace of the construction rules) for the list $[3, 4, [7, [8, 2], 9], 122, [2, 2]]$. 25pt

Solution: We choose the abstract data type

$$\langle\{\mathbb{N}, \mathbb{L}\}, \{[c_l \colon \mathbb{L} \times \mathbb{L} \to \mathbb{L}], [c_n \colon \mathbb{N} \times \mathbb{L} \to \mathbb{L}], [nil \colon \mathbb{L}], [o \colon \mathbb{N}], [s \colon \mathbb{N} \to \mathbb{N}]\}\rangle$$

The constructors c_l and c_l construct lists by adding a list or a number at the front of the list. With this, the list above has the constructor term.

$$c_n(\underline{3}, c_n(\underline{4}, c_l(c_n(\underline{7}, c_l(c_n(\underline{8}, c_n(\underline{2}, nil)), c_n(\underline{9}, nil)), c_n(\underline{122}), c_l(c_n(\underline{2}, c_n(\underline{2}, nil)))nil))))$$

where $\underline{n}$ is the s, o-constructor term of the number n.

6 Assignment 6 (Abstract Datatypes and Abstract Procedures) – Given Oct. 19., Due Oct. 26.

Problem 6.1 (Abstract Procedures)
Given the ADT for binary numbers (think of them as lists of 0 and 1 digits): 25pt

$$\langle\{\mathbb{D},\mathbb{B}\},\{[0\colon\mathbb{D}],[1\colon\mathbb{D}],[bit\colon\mathbb{D}\to\mathbb{B}],[addBit\colon\mathbb{B}\times\mathbb{D}\to\mathbb{B}]\}\rangle$$

and the binary operator XOR

XOR	1	0
1	0	1
0	1	0

The bitwise XOR (written as $\oplus$) of two binary numbers performs the logical XOR operation on each pair of corresponding digits (bits). Example:

$$\begin{array}{rr} & 0110 \\ \oplus & 1010 \\ \hline & 1100 \end{array}$$

1. Write down an abstract procedure $\oplus$ that
   - computes the bitwise XOR of two binary numbers if they have equal lengths (leading zeros are accepted)
   - does not terminate otherwise
2. Represent the two numbers 101 and 011 using the ADT above and show the computation process of $\oplus$ on these two numbers.

Solution:

1. the following procedures are one possible solution:
$$\langle\chi{::}\mathbb{D}\times\mathbb{D}\to\mathbb{D}\,;\,\{\chi(0,0)\rightsquigarrow 0,\chi(0,1)\rightsquigarrow 1,\chi(1,0)\rightsquigarrow 1,\chi(1,1)\rightsquigarrow 0\}\rangle$$
$$\langle\oplus{::}\mathbb{B}\times\mathbb{B}\to\mathbb{B}\,;\,\{\begin{array}{l}\oplus(\mathrm{bit}(x),\mathrm{bit}(y))\rightsquigarrow\mathrm{bit}(\chi(x,y))\\ \oplus(\mathrm{bit}(x),\mathrm{addBit}(Y,y))\rightsquigarrow\oplus(\mathrm{bit}(x),\mathrm{addBit}(\mathrm{addBit}(Y,y),1))\\ \oplus(\mathrm{addBit}(X,x),\mathrm{bit}(y))\rightsquigarrow\oplus(\mathrm{addBit}(\mathrm{addBit}(X,x),1),\mathrm{bit}(y))\\ \oplus(\mathrm{addBit}(X,x),\mathrm{addBit}(Y,y))\rightsquigarrow\mathrm{addBit}(\oplus(X,Y),\chi(x,y))\end{array}\}\rangle$$
2. The numbers are represented as $\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(1),0),1)$ and $\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(0),1),1)$:
$$\begin{array}{l}\oplus(\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(1),0),1),\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(0),1),1))\\ \rightsquigarrow\mathrm{addBit}(\oplus(\mathrm{addBit}(\mathrm{bit}(1),0),\mathrm{addBit}(\mathrm{bit}(0),1)),\chi(1,1))\\ \rightsquigarrow\mathrm{addBit}(\mathrm{addBit}(\oplus(\mathrm{bit}(1),\mathrm{bit}(0)),\chi(0,1)),0)\\ \rightsquigarrow\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(\chi(1,0)),1),0)\\ \rightsquigarrow\mathrm{addBit}(\mathrm{addBit}(\mathrm{bit}(1),1),0)\end{array}$$

Problem 6.2 (Angry Birds)
Your new favorite game is Angry Birds, and, after a lazy afternoon when you have played the game, you observed the following rules for deducing the score: 25pt

- the red bird will always add 5000 points to your score (no matter what it hits)
- the blue bird always is split into 3 smaller birds, and every bird adds 1000 points if it hits an object
- the yellow bird will add 2500 birds only if it hits a green pig
- the white bird is considered peaceful and will add no points to your score

Your version of Angry Birds permits you to choose K from a list of birds which you will fire. Therefore, you design the following SML datatype:

```
datatype angrybird = white | red | blue of int | yellow of bool;
```

The int parameter of the blue bird tells you how many birds hit an object, and the bool parameter of the yellow bird tells you if the bird hits a pig or not.

You are required to write an SML function getScore which takes a list of Angry Birds and the number K of birds you can fire and returns the maximum score you can get by firing your choice of K birds:

```
val getScore = fn : angrybird list * int -> int;
- getScore( [white, white, red, blue(3)], 3 );
val it = 8000 : int; (* red, blue and one of the whites are chosen *)
- getScore( [red, yellow(false), yellow(true), blue(2)], 2);
val it = 7500 : int; (* red and yellow(true) are chosen *)
```

Solution:

```
datatype angrybird = white | red | blue of int | yellow of bool;

fun convert(white) = 0
  | convert(red) = 5000
  | convert(blue(x)) = x*1000
  | convert(yellow(x)) = if x then 2500 else 0;

fun getScore(l,k)=
let
val l1 = map convert l
val l2 = ListMergeSort.sort (op<) l1
in
foldl (op+) 0 (List.take(l2, k))
end;
```

Problem 6.3 (Abstract Procedures on a Deck of Cards)
You are given the following abstract data type for a deck of cards: 35pt

$$\langle \{\mathbb{D}, \mathbb{C}, \mathbb{S}, \mathbb{N}, \mathbb{B}\}, \{[nil\colon \mathbb{D}], [S\colon \mathbb{S}], [H\colon \mathbb{S}], [C\colon \mathbb{S}], [D\colon \mathbb{S}], [o\colon \mathbb{N}], [s\colon \mathbb{N} \to \mathbb{N}], [card\colon \mathbb{S} \times \mathbb{N} \to \mathbb{C}], [add\colon \mathbb{C} \times \mathbb{D} \to \mathbb{D}], [T\colon \mathbb{B}], [F\colon \mathbb{B}]$$

1. given a deck, count the number of cards of a specific suit:

$$count\_suit(add(card(S, s(s(o))), add(card(D, s(s(s(o)))), add(card(S, s(s(s(o)))), nil))), S)$$
$$= s(s(o))$$

2. given a deck, count the number of cards with a specific number:

$$count\_number(add(card(S, s(s(o))), add(card(D, s(s(s(o)))), add(card(S, s(s(s(o)))), nil))),$$
$$s(s(s(o)))) = s(s(o))$$

3. given a deck, reverse the order of the cards inside:

$$reverse(add(card(H, s(s(o))), add(card(D, s(s(s(o)))), add(card(S, s(s(s(o)))), nil))))$$
$$= add(card(S, s(s(s(o)))), add(card(D, s(s(s(o)))), add(card(H, s(s(o)))), nil)$$

4. given 2 decks, check if the first is included in the second:

$$included(add(card(D, s(s(o))), add(card(S, s(o)), nil)),$$
$$add(card(D, s(s(o))), add(card(H, s(s(s(o)))), add(card(S, s(o)), nil)))) = T$$

5. given 2 decks, check if they contain they contain the same cards:

$$same(add(card(D, s(s(o))), add(card(S, s(s(o))), nil)), add(card(D, s(s(o))),$$
$$add(card(H, s(s(o))), nil))) = F$$

6. given a deck, sort the cards inside so that they are first ordered by suit (spades, hearts, diamonds and then clubs) and then in increasing order by their number:

$$sort(add(card(D, s(s(o))), add(card(H, s(s(o))), add(card(D, s(s(s(o)))),$$
$$add(card(S, s(s(o))), add(card(C, s(s(s(o)))), nil))))))$$
$$= add(card(S, s(s(o))), add(card(H, s(s(o))), add(card(D, s(s(o))),$$
$$add(card(D, s(s(s(o)))), add(card(C, s(s(s(o)))), nil)))))$$

Solution:

Problem 6.4 (Substitutions) 15pt

Let $\langle\{\mathbb{A}\}, \{[f\colon \mathbb{A} \times \mathbb{A} \times \mathbb{A} \to \mathbb{A}], [g\colon \mathbb{A} \to \mathbb{A}], [a\colon \mathbb{A}]\}\rangle$ be an abstract datatype and

$$s := f(g(x_{\mathbb{A}}), y_{\mathbb{A}}, f(x_{\mathbb{A}}, h, x_{\mathbb{A}})), \quad t := f(g(g(a)), h, f(g(a), h, g(a)))$$

be two constructor terms of sort $\mathbb{A}$.

1. Find a substitution σ such that $\sigma(s) = t$
2. Compute $\sigma(u)$, where $u := f(g(y_{\mathbb{A}}), z_{\mathbb{A}}, x_{\mathbb{A}})$

Solution:

1. $\sigma := [g(a)/x_{\mathbb{A}}], [h/y_{\mathbb{A}}]$
2. $f(g(h), z_{\mathbb{A}}, g(a))$

7 Assignment 7 (More SML) – Given Oct. 27., Due Nov. 9.

Problem 7.1 (Mutual recursion in SML)
Implement functions for the following recursive/ mutually recursive functions: 20pt

- the Hofstadter H sequence:

$$H(n) = \begin{cases} 0 & \text{if } n = 0 \\ n - H(H(H(n-1))) & \text{if } n>0 \end{cases}$$

- the Hofstadter Q sequence:

$$Q(n) = \begin{cases} 1 & \text{if } n = 1 or n = 2 \\ Q(n - Q(n-1)) + Q(n - Q(n-2)) & \text{if } n>2 \end{cases}$$

- the Hofstadter male and female sequences:

$$\begin{aligned} male(n) &= \begin{cases} 0 & \text{if } n = 0 \\ n - female(male(n-1)) & \text{if } n>0 \end{cases} \\ female(n) &= \begin{cases} 1 & \text{if } n = 0 \\ n - male(female(n-1)) & \text{if } n>0 \end{cases} \end{aligned}$$

- the following mutually recursive functions:

$$\begin{aligned} a(n) &= \begin{cases} 1 & \text{if } n = 0 \\ 2 & \text{if } n = 1 \\ a(n-2) * Q(n+1) & \text{if } n>1 \end{cases} \\ b(n) &= \begin{cases} c(1) & \text{if } n = 0 \\ c(2) & \text{if } n = 1 \\ c(n-2) & \text{if } n>1 \end{cases} \\ c(n) &= \begin{cases} 1 & \text{if } n = 0 \\ a(n-1) * c(n-1) & \text{if } n>0 \end{cases} \\ d(n) &= \begin{cases} 0 & \text{if } n = 0 \\ b(a(n)) & \text{if } n>0 \end{cases} \end{aligned}$$

Please raise appropriate exceptions when the arguments of the functions are negative.

Solution:

```
exception Negative;

(* Hofstadter H *)
fun H(0) = 0
  | H(n) = if n < 0 then raise Negative else n - H(H(H(n-1)));

(* Hofstadter Q *)
```

```
fun Q(1) = 1
  | Q(2) = 1
  | Q(n) = if n < 1 then raise Negative else Q(n−Q(n−1)) + Q(n−Q(n−2));

(* Hofstadter male & female *)
fun female(0) = 1
  | female(n) = if n < 0 then raise Negative else n − male(female(n−1))
and male(0) = 0
  | male(n) = if n < 0 then raise Negative else n − female(male(n−1));

(* random mutual recursion *)
fun a(0) = 1
  | a(1) = 2
  | a(n) = if n < 0 then raise Negative else a(n−2) * Q(n+1)
and b(0) = c(1)
  | b(1) = c(2)
  | b(n) = if n < 0 then raise Negative else c(n−2)
and c(0) = 1
  | c(n) = if n < 0 then raise Negative else a(n−1) * c(n−1)
and d(0) = 0
  | d(n) = if n < 0 then raise Negative else b(a(n));
```

Problem 7.2 (Second Maximum)

Write a function that takes a list of real numbers and returns the second maximum element of the list and raise an exception if there is no such element. For example, max([1.0, 4.4, 4.5, 6.2, 2.3, 2.2, 2.2]) should return 4.5. 20pt

---

Solution:

```
exception WrongInput;
fun selmax(nil: real list) = nil
| selmax(a::nil) = a::nil
| selmax(a::b::t) = if a>b then b::selmax(a::t) else a::selmax(b::t);
fun remlast(nil) = raise WrongInput
| remlast(l) = rev(tl(rev(l)));
fun max(nil) = raise WrongInput
| max(h::nil) = raise WrongInput
| max(l) = hd(rev(selmax(remlast(selmax(l)))));
```

Problem 7.3 (Implementing a simple grep)

Your task is to implement a function that works similarly to the grep functionality in Linux. The input contains three strings: a pattern string, an input file name, and an output file name. You have to implement a function grep(pattern,infile,outfile), which writes to the output file those lines from the input file, which contain the pattern string. 30pt

For example: Given pattern substring:we have

input	output
This line contains a substring. This line contains strings. Strings contain substrings. Substrings are strings.	This line contains a substring. Strings contain substrings.

Solution:

```
fun readFile(filename) =
  let
    val fin = TextIO.openIn(filename)
    fun read() =
      let
        val x = TextIO.inputLine(fin)
      in
        if x = NONE then [] else (valOf x) :: read()
      end
    in
      read()
    end;

fun writeFile(filename,data) =
  let
    val fout = TextIO.openOut(filename)
    fun write([]) = (TextIO.flushOut(fout); true)
      | write(l::ls) = (TextIO.output(fout,l); write(ls))
    in
      write(data)
    end;

fun grepList(_,nil) = nil
  | grepList(pattern,l::ls) = if (String.isSubstring pattern l)
        then l::grepList(pattern,ls)
        else grepList(pattern,ls);

fun grep(pattern,infile,outfile) = writeFile(outfile,grepList(pattern,readFile(infile)));
```

Problem 7.4 (ASCII code)

Is the `ASCII` code a prefix code? Explain. Write down the statement for each of the theorems you are using. 20pt

Solution: ASCII is a prefix code because it assigns a number between 0 and 127 to each letter (character). In binary, each character gets a sequence of 8 (7 also accepted) bits, and as long as the codes for every character have the same length, the code is a prefix code.

Theorem used in demonstration: 5.30

"If c is a code with $|c(a)| = k$ for all $a \in A$ for some $k \in \mathbb{N}$ then c is a prefix code."

Problem 7.5 Let $A := \{a, b, c, d, e, f, g, h\}$ and $\mathbb{B} := \{0, 1\}$, and 20pt

$c(a)$:=010010010101001	$c(b)$:=010110010101001
$c(c)$:=010011110101001	$c(d)$:=010010011101001
$c(e)$:=010010010110001	$c(f)$:=010010010101101
$c(g)$:=010011110101000	$c(h)$:=011111110101000

Is c a character code? Does it induce a code on strings? Justify your answers to both questions.

Problem 7.6 (Testing for prefix codes)

Write an SML function prefix_code that tests whether a code is a prefix code. The code is given as a list of pairs (SML type char∗string list). 20pt

Example:

```
prefix_code [(#"a","0"), (#"b","1")];
val it = true : bool
```

Solution:

```
infix mem (* list membership *)
fun x mem nil = false | x mem (y::l) = (x=y) orelse (x mem l)
(* test for repeated elements in list *)
fun repeat nil = false
  | repeat (h::t) = h mem t orelse repeat(t)
fun function rel = not (repeat (map (fn (x,_) => x ) rel))
fun injective rel = not (repeat (map (fn (_,x) => x ) rel))
(* test whether a list is a prefix of another *)
fun prefix_list _ nil = false
  | prefix_list nil _ = true
  | prefix_list (h::t) (k::l) = if (h = k) then prefix_list t l else false;
(* testing if there is an element with property p in list *)
fun exists p nil = false | exists p (h::t) = p h orelse exists p t;
(* testing for the prefix property *)
fun prefix_prop code =
      exists (fn (_,c) =>
                exists (fn (_,d) =>
                          prefix_list (explode c) (explode d))
                       code)
                code;
(* putting it all together *)
fun prefix_code code = function code
    andalso injective code
    andalso prefix_prop code = false;


(*Test cases:*)
val test1 = prefix_code [(#"a","0"), (#"b","10")] = true;
val test2 = prefix_code [(#"a","0"), (#"b","1")] = true;
val test3 = prefix_code [(#"a","0"), (#"b","10"), (#"c", "110")]=true;
val test4 = prefix_code [(#"a","0"), (#"a","10")]=false;
val test5 = prefix_code [(#"a", "0"), (#"b", "01")]=false;
```

```
val test6 = prefix_code [(#"a", "10"), (#"b", "101"), (#"c", "01")]=false;
val test7 = prefix_code [(#"a", "10"), (#"b", "11")]=true;
val test8 = prefix_code [(#"a","0")]=true;
val test9 = prefix_code []=true;
```

Problem 7.7 (Codes)

Given the following mapping from characters to binary strings: 20pt

a = 00001	b = 01010	c = 10100	d = 00100	e = 1110	f = 01001
g = 00101	h = 1001	i = 10111	j = 01111	k = 10101	l = 00000
m = 01100	n = 01000	o = 110	p = 01101	q = 10110	r = 11111
s = 00010	t = 1000	u = 11110	v = 00011	w = 01011	x = 01110
y = 00111	z = 00110				

1. Is this a character code?
2. Is this a prefix code?
3. Decode the string 010001101000010100000100100 into the english alphabet.

For the first two tasks say what you are checking and restrict yourselves to the characters a to l.

Solution:

1. Yes, it is a valid code since every letter gets mapped to a different one.
2. Yes, it is a prefix code since no character is a prefix of any other one.
3. The decoded string is "notbad"

---

8 Assignment 8 (Landau Sets and Quine McCluskey Algorithm) – Given Nov. 16. 2016

Problem 8.1 (Manual QMC application)
Manually execute the Quine-McCluskey algorithm to get the minimum polynomial for the function with the provided truth table: 25pt

x_0	x_1	x_2	x_3	f
F	F	F	F	T
F	F	F	T	F
F	F	T	F	F
F	F	T	T	T
F	T	F	F	F
F	T	F	T	F
F	T	T	F	T
F	T	T	T	T
T	F	F	F	F
T	F	F	T	T
T	F	T	F	F
T	F	T	T	T
T	T	F	F	T
T	T	F	T	F
T	T	T	F	T
T	T	T	T	F

Problem 8.2 (Landau Sets)
Determine for the following functions f and g whether $f \in O(g)$, or $f \in \Omega(g)$, or $f \in \Theta(g)$, explain your answers. 25pt

f	g	f	g
n^4	$n^3 \log n$	$\frac{n^2}{\log n}$	$n^4 + \log n$
$n + n^2$	$n + 3n^3$	$\ln n$	$\log_{10}(n)$
e^{n^2}	2^n	5^n	$n!$
n	$\sin n$	9^n	$n^{\log_n(8^n)}$
12	42	$(n^3+3)^2$	$(3n^3+2)^2$

Problem 8.3 (Implementing Quine-McCluskey)
Implement the Quine-McCluskey algorithm in SML. The function QuineMcCluskey should take a truth table of type (int list $*$ int) list as input and return a minimal covering prime implicant polynomial as a string. 50pt

Please follow the steps described in the slides and mark them clearly in your code. Verify your solution for the first problem using this algorithm.

Example:

```
- QuineMcCluskey([([0,0], 1),([0,1], 0),([1,0], 0),([1,1], 1)])
val it = "(-x1)(-x2) + x1x2" : string
```

9 Assignment 9 (Propositional Logic) – Given Nov. 23., Due Nov. 30.

Problem 9.1 (Propositional Logic with Hilbert Calculus) 30pt

- Give an example for each of the following:
  1. a Boolean expression which is falsifiable, but also satisfiable
  2. a Boolean expression which is valid
  3. a Boolean expression which is unsatisfiable
- Consider the Hilbert-style calculus given by the axioms and inference rules:
  1. $K := P \Rightarrow Q \Rightarrow P$
  2. $S := (P \Rightarrow Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$
  3. $\dfrac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{A}}{\mathbf{B}}$ MP
  4. $\dfrac{\mathbf{A}}{[\mathbf{B}/X](\mathbf{A})}$ Subst

  Prove the formula $\mathbf{A} \Rightarrow \mathbf{C} \Rightarrow \mathbf{C}$ in this calculus.

Problem 9.2 (Alternative Calculus)

Consider a calculus given by the axioms $\mathbf{A} \vee \neg\, \mathbf{A}$ and $\mathbf{A} \wedge \mathbf{B} \Rightarrow \mathbf{B} \wedge \mathbf{A}$ and the following rules: 20pt

$$\frac{\mathbf{A} \Rightarrow \mathbf{B}}{\neg\, \mathbf{B} \Rightarrow \neg\, \mathbf{A}}\ Transp \qquad \frac{\mathbf{A}}{[\mathbf{B}/P]\mathbf{A}}\ \text{Subst}$$

Prove that the calculus is sound.

Problem 9.3 (Check properties of Boolean formulae in SML)

Given the following SML data types for Boolean formulae 50pt

```
datatype boolexp = zero | one
                 | plus of boolexp * boolexp
                 | times of boolexp * boolexp
                 | compl of boolexp
                 | var of int
```

write the following SML functions of type boolexp -> int

valid	returns 1 if expression is valid and 0 otherwise
satisfiable	returns 1 if expression is satisfiable and 0 otherwise
unsatisfiable	returns 1 if expression is unsatisfiable and 0 otherwise
falsifiable	returns 1 if expression is falsifiable and 0 otherwise

Examples:

```
- valid(plus(var(1),compl(var(1))));
val it = 1 : int
- falsifiable(plus(var(1),one));
```

```
val it = 0 : int
```